

# Aufgabe 10

Gruppe 01

3 7 2020

Laden sie den Workspace yingtang\_20ueb9.RData.

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v ggplot2 3.3.0      v purrr   0.3.4
## v tibble  3.0.1      v dplyr  0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

load("data/yingtang_20_ueb9.Rdata")

# Koordinatensystem anpassen
library(sp)
proj4string(terrain) <- CRS("+proj=utm +zone=50 +ellps=WGS84 +datum=WGS84")
proj4string(ljz) <- proj4string(terrain)
```

## Aufgabe 22. Inverse Distance Weighting

Deterministische Modelle sind bei räumlichen Interpolationsfragen äußerst beliebt, weil sie verhältnismäßig leicht angewendet werden können. Ähnlich der meisten räumlichen Prognosemodelle basiert auch das Inverse Distance Weighting (IDW) auf einem gewichteten Mittel der vorhandenen Messwerte. Allerdings beruht die Berechnung der Gewichte nicht auf stochastischen Überlegungen, sondern einzig auf der subjektiven Wahl eines Parameters.

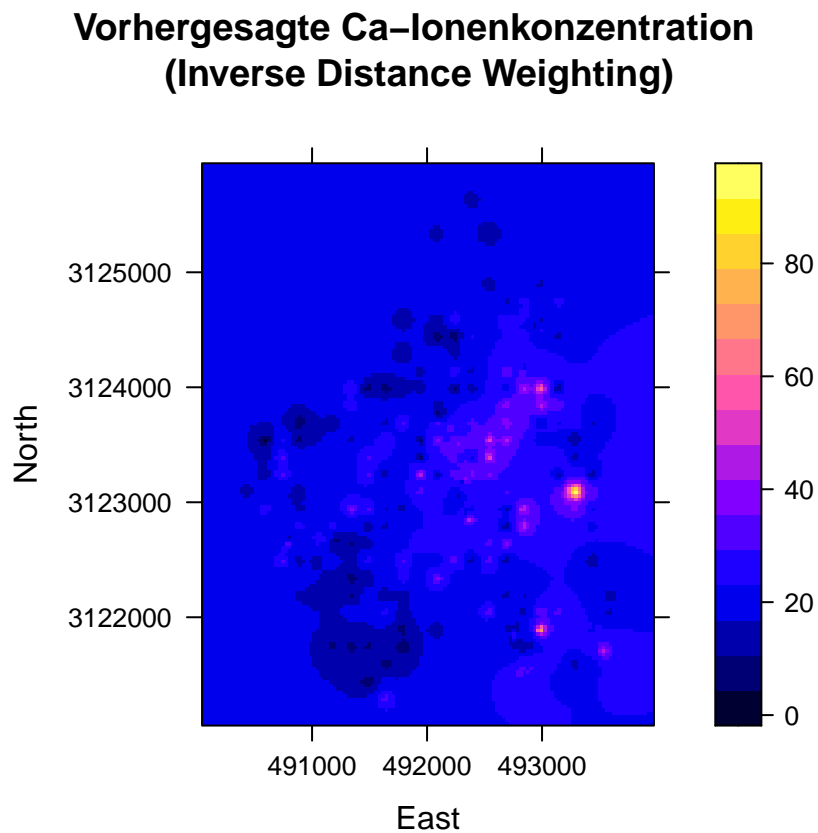
- a) Laden Sie den oben genannten Workspace in Ihrem Markdown Skript und führen Sie mit Hilfe der kriging-Funktion eine IDW-Vorhersage der Ca<sup>2+</sup>-Ionen durch. Nutzen Sie das Objekt „terrain“ des geladenen Workspace als Ziel-Grid und plotten Sie ihr Ergebnis in gewohnt aussagekräftiger Manier. Ermitteln Sie außerdem den RMSE dieser Methode, indem Sie eine LOOK-Kreuzvalidierung durchführen. (2 Punkte)

```
#Inverse-Distance-Weighting
library(gstat)
IDW <- kriging(Ca_exch~1,
               locations = ljz,
               newdata = terrain)

## [inverse distance weighted interpolation]

#Darstellung
library(lattice)
```

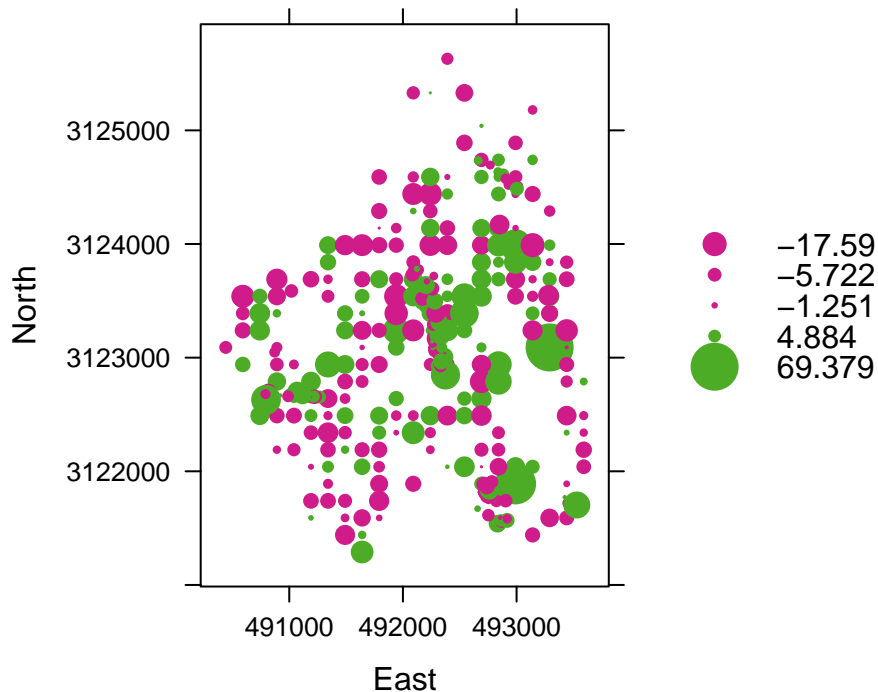
```
spplot(IDW, "var1.pred",
       main = "Vorhergesagte Ca-Ionenkonzentration\n (Inverse Distance Weighting)",
       scales=list(tick.number=4, alternating=1),
       xlab="East",
       ylab="North")
```



```
#Leave-one-out-Cross-Validation
idw_L00CV <- gstat::krige.cv(Ca_exch ~ 1,
                             ljz)

bubble(idw_L00CV, "residual",
       main = "Residuen der L00CV-Analyse\n (Inverse Distance Weighting)",
       scales=list(tick.number=4, alternating=1),
       xlab="East",
       ylab="North")
```

## Residuen der LOOCV-Analyse (Inverse Distance Weighting)



```
# RMSE
rmse <- function(x,y) {
  sqrt(mean((x-y)^2))
}

idw_rmse <- rmse(x = idw_LOOCV$var1.pred,
  y = idw_LOOCV$observed)
```

### Aufgabe 23. Kriging with External Drift

Der Ansatz des Ordinary Kriging (OK) berücksichtigt zwar autokorrelatives Verhalten der Zielgröße im Raum, deterministische Trends bleiben jedoch außen vor. Das Verfahren der Multiplen Linearen Regression (MLR) versucht die abhängige Variable anhand additiv zusammenwirkender Kovariablen abzubilden ohne jedoch mögliche räumliche Zusammenhänge innerhalb der Zielgröße zu verarbeiten. Eine verbreitete Methode, die beide Aspekte kombiniert, ist das Regression Kriging (RK) oder auch Kriging with External Drift (KED).

- a) Leiten Sie für die Residuen der Regression aus Aufgabe 21 ein geeignetes Variogrammodell ab. Vergleichen Sie die Charakteristik dieses Modells mit dem autokorrelativen Verhalten der Original-Zielgröße (s. Aufgabe 12). Erklären Sie die Unterschiede in knappen Worten. (2 Punkte)

```
#Variogram aus dem Regressionsmodell aus 21
lr_variomodel <- variogram(Ca_exch ~ yingtang_elevation + SAGAWI,
  data = l1jz,
  cutoff = 2202,
  width = 150)

m <- vgm(model = "Exp",
```

```

        cutoff = 2202)

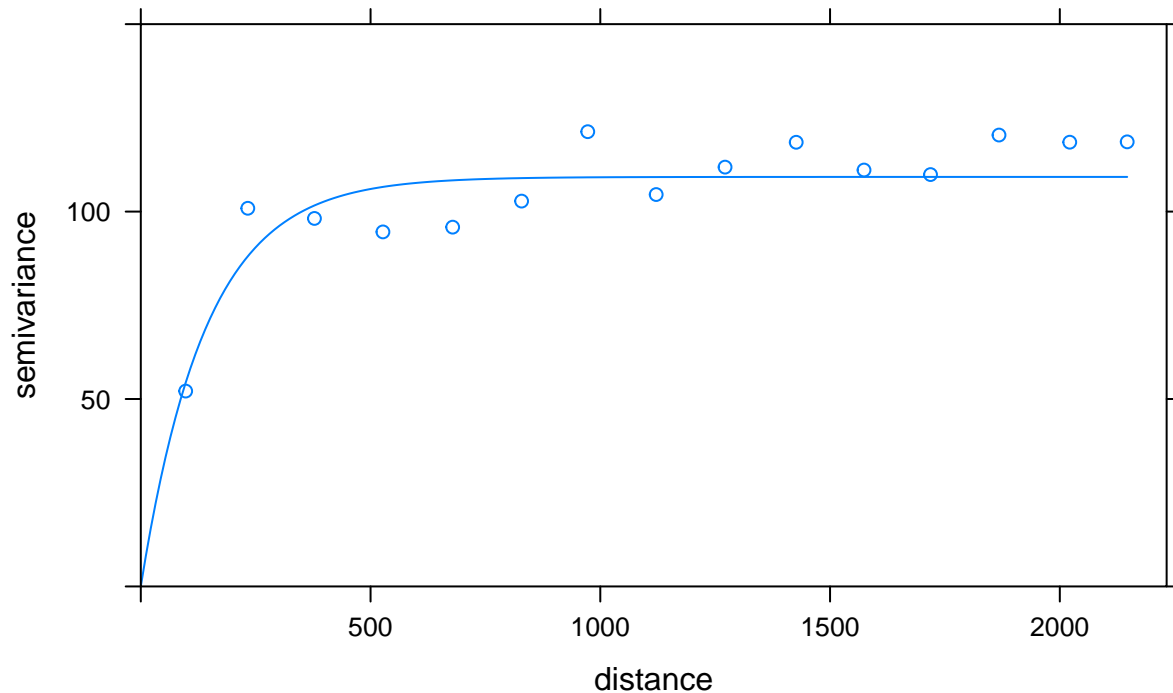
lr_fvariomodel <- fit.variogram(lr_variomodel,
                               model = m,
                               fit.method = 7)

lr_fvariomodel

##   model    psill    range
## 1   Exp 109.2444 141.8968
plot(lr_variomodel,
     model = lr_fvariomodel,
     cutoff = 2202,
     ylim = c(0, 150),
     main = "Variogrammmodell der austauschbaren Ca-Ionen\n(lineare Regression)")

```

### Variogrammmodell der austauschbaren Ca-Ionen (lineare Regression)



Modell bezogen auf die räumliche Nähe sill: 117.75 range: 293.96

Lineares Regressionsmodell sill: 109.24 range: 141.89 -> Orte, die einen sehr ähnlichen SAGAWI und eine sehr ähnliche Höhenlage haben, ähneln sich auch stark im Wert für die austauschbaren Ca-Ionen. Da die range noch kleiner ist, als bei dem Modell aus Aufgabe 12, müssen die Werte des Höhenmodells und des SAGAWI vom Probeentnahmeort und der zu berechneten Stelle nahezu gleich sein, um eine geeignete Aussage über den Ca<sub>exch</sub> Wert treffen zu können.

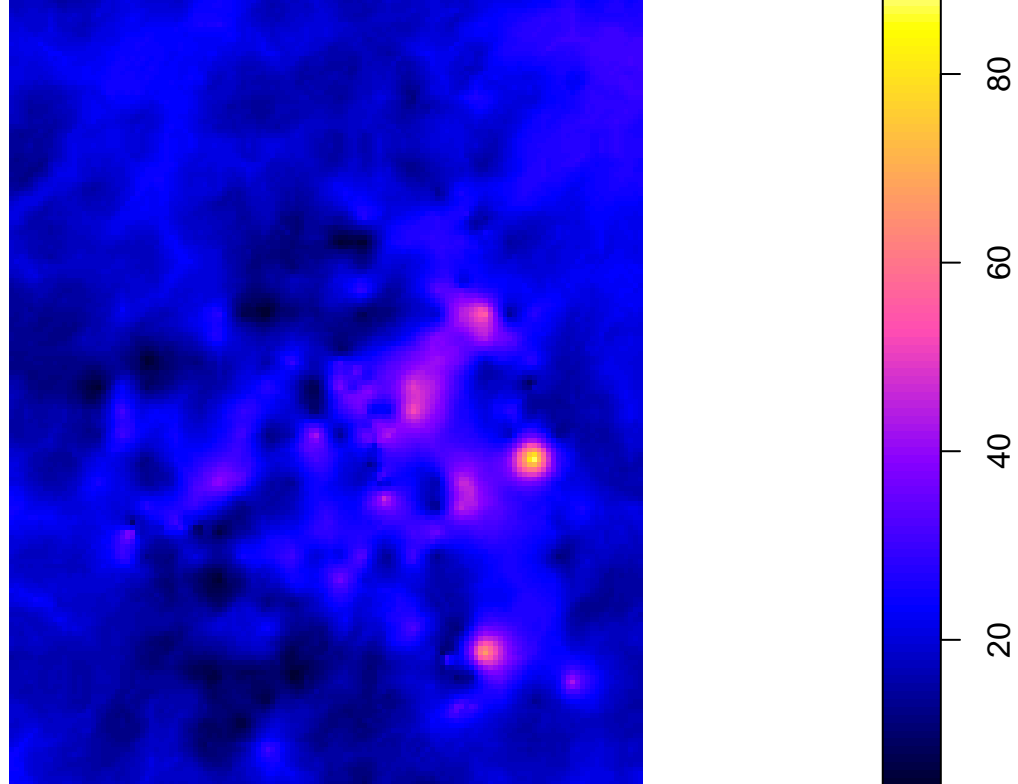
- b) Wenden Sie die Methode krige entsprechend des KED an und interpolieren Sie die Konzentration der Ca-Ionen auf die Rasterzellen des Objekts „terrain“. Ermitteln Sie anschließend den RMSE dieser Methode. (2 Punkte)

```
#Universal Kriging = KED
uk <- krige(Ca_exch ~ yingtang_elevation + SAGAWI,
            ljz,
            terrain,
            model = lr_fvariomodel)
```

```
## [using universal kriging]
```

```
plot(uk,
      main = "Vorhersage durch Universal Kriging")
```

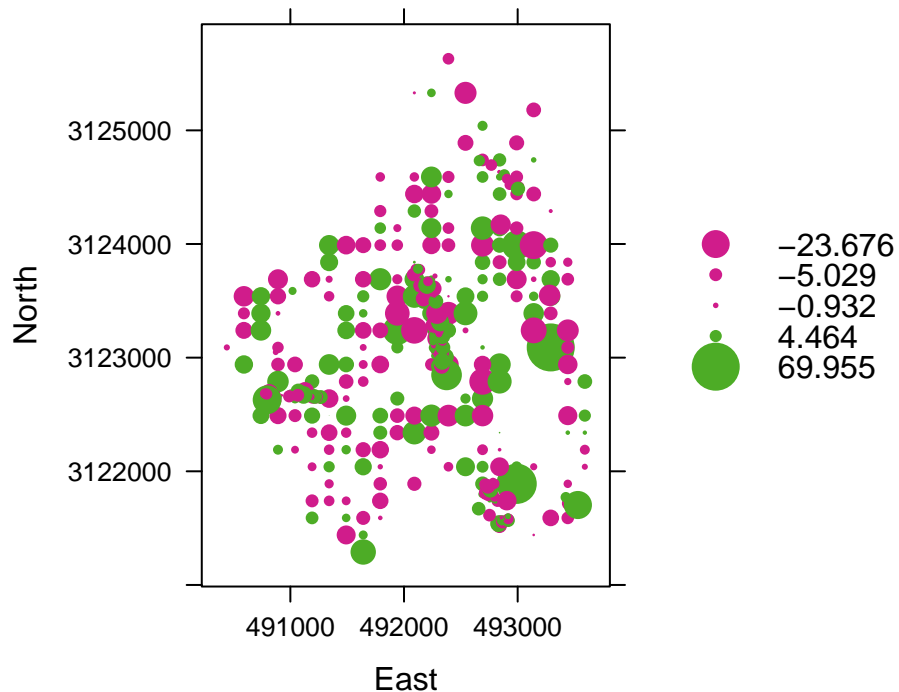
### Vorhersage durch Universal Kriging



```
#Leave-one-out-Cross-Validation
ukLOOCV <- gstat::krige.cv(Ca_exch ~ yingtang_elevation + SAGAWI,
                           ljz,
                           model = lr_fvariomodel)

bubble(ukLOOCV, "residual",
        main = "Residuen der LOOCV-Analyse\n (Universal Kriging)",
        scales=list(tick.number=4,
                     alternating=1),
        xlab="East",
        ylab="North")
```

## Residuen der LOOCV-Analyse (Universal Kriging)



```
#RMSE
uk_rmse <- rmse(x = ukLOOCV$var1.pred,
               y = ukLOOCV$observed)
```

- c) Vergleichen Sie das KED-Ergebnis mit den Resultaten von MLR, OK und IDW. Für welches Interpolationsverfahren würden Sie sich entscheiden? Begründen Sie ihre Wahl in ein, zwei Sätzen. (3 Punkte)

```
##Multiple linear Regression##
mlr <- krige(Ca_exch ~ yingtang_elevation + SAGAWI,
            ljz,
            terrain)
```

```
## [ordinary or weighted least squares prediction]
```

```
#Leave-one-out-Cross-Validation
mlrLOOCV <- gstat::krige.cv(Ca_exch ~ yingtang_elevation + SAGAWI,
                          ljz)
```

```
#RMSE
mlr_rmse <- rmse(x = mlrLOOCV$var1.pred,
               y = mlrLOOCV$observed)
```

```
##Ordinary Kriging##
#Variogramm Ordinary Kriging
ok_variogram <- variogram(Ca_exch ~ EAST + NORTH,
                        data = ljz,
                        alpha = c(0, 45, 90, 135),
```

```

        cutoff = 4400,
        width = 150)
#Modell zum Variogramm
ok_fvariogram <- fit.variogram(ok_variogram,
                             vgm(model = "Sph"))
#Ordinary Kriging
ok <- gstat::krige(Ca_exch ~ 1,
                  ljz, terrain,
                  model = ok_fvariogram)

## [using ordinary kriging]
#Leave-one-out-Cross-Validation
okLOOCV <- gstat::krige.cv(Ca_exch ~ yingtang_elevation + SAGAWI,
                          ljz,
                          model = ok_fvariogram)

#RMSE
ok_rmse <- rmse(x = okLOOCV$var1.pred,
                y = okLOOCV$observed)

##Inverse Distance weighting##
IDW <- krige(Ca_exch ~ 1,
             locations = ljz,
             newdata = terrain)

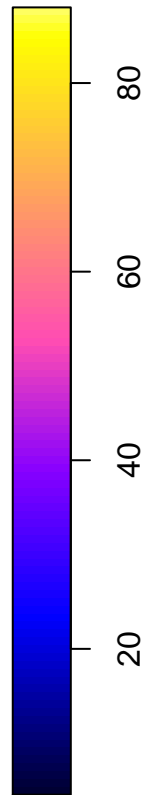
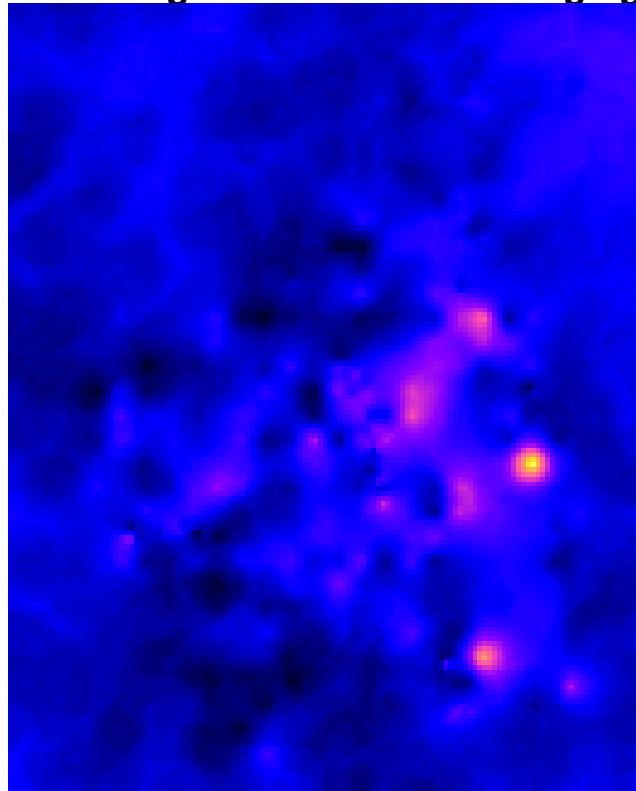
## [inverse distance weighted interpolation]
#Leave-one-out-Cross-Validation
idwLOOCV <- gstat::krige.cv(Ca_exch ~ yingtang_elevation + SAGAWI,
                           ljz)

#RMSE
idw_rmse <- rmse(x = idwLOOCV$var1.pred,
                 y = idwLOOCV$observed)

#Vergleichsplot der Vorhersagen
plot(uk,
     main = "Vorhersage durch Universal Kriging")

```

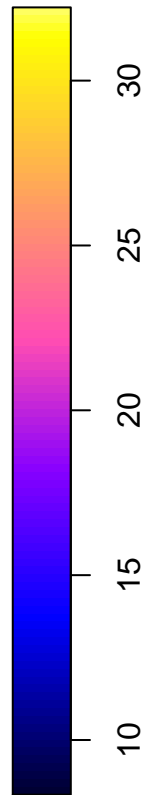
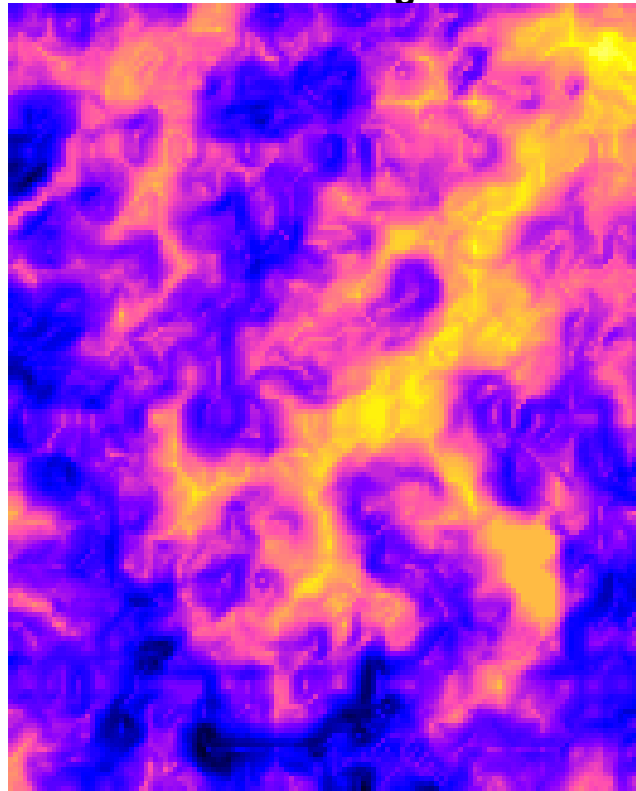
## Vorhersage durch Universal Kriging



```
plot(mlr,  
      main = "Residuen des linearen Regressionsmodells")
```

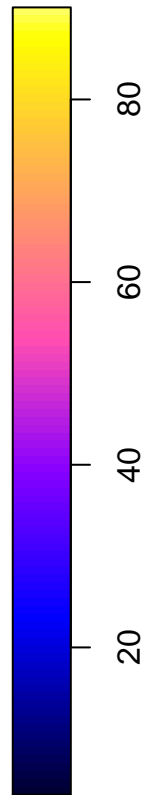
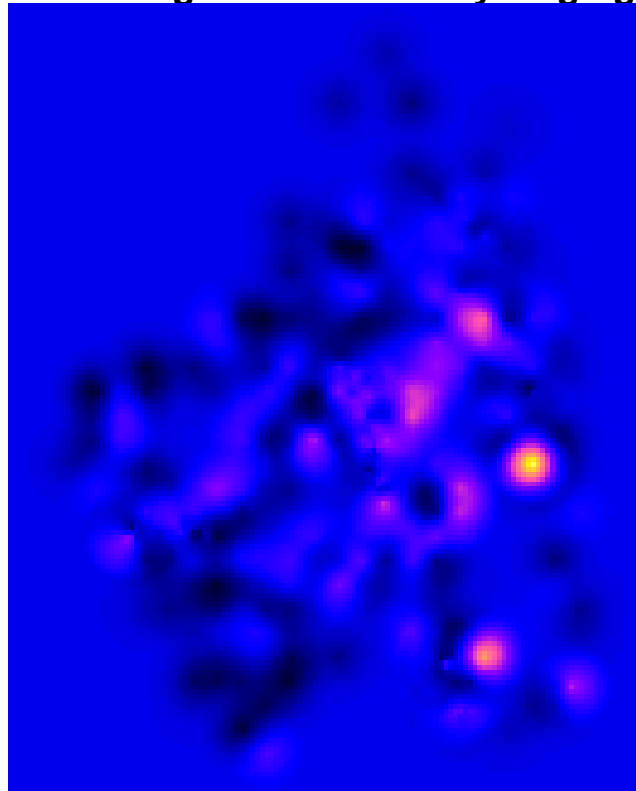


## Residuen des linearen Regressionsmodells



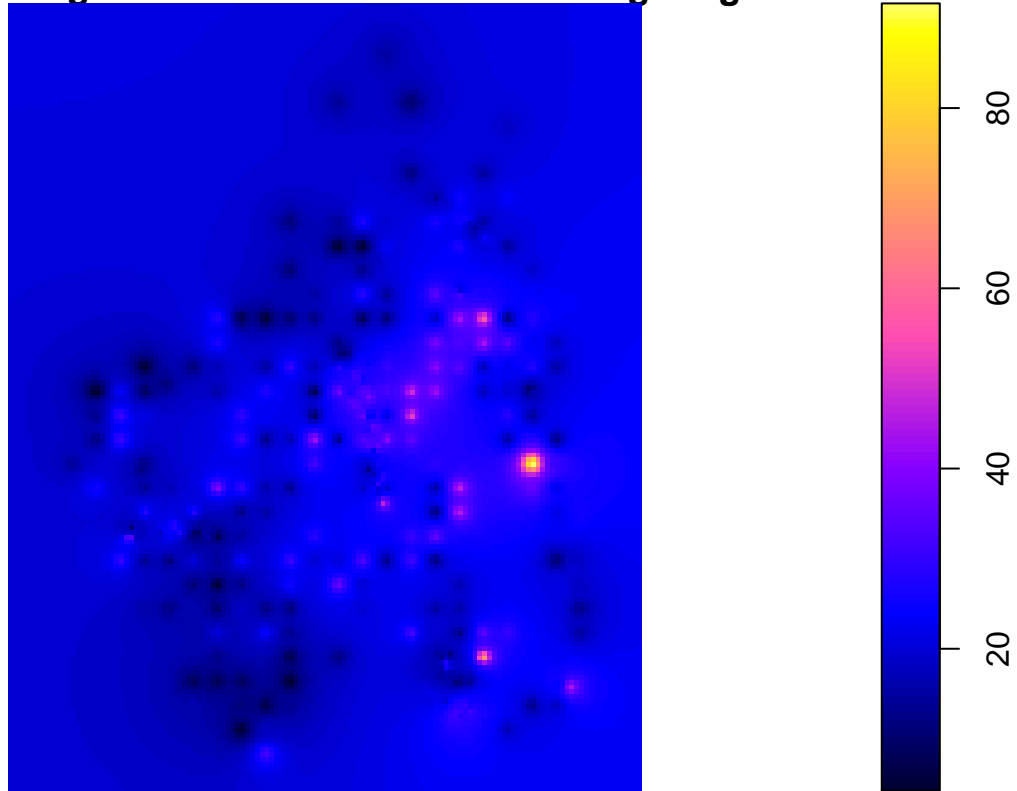
```
plot(ok,  
      main = "Vorhersage durch Ordinary Kriging")
```

## Vorhersage durch Ordinary Kriging



```
plot(IDW,  
     main = "Vorhersage durch Inverse-Distance-Weighting")
```

## Vorhersage durch Inverse-Distance-Weighting



```
#RMSEs  
idw_rmse
```

```
## [1] 10.41291
```

```
uk_rmse
```

```
## [1] 9.29913
```

```
ok_rmse
```

```
## [1] 9.305209
```

```
mlr_rmse
```

```
## [1] 10.41291
```

Der *RMSE* ist je für das *OK* und *KED* (9,3; 9,299), sowie für das *IDW* und das *Kriging* mittels des *linearen Regressionsmodells* (10,41) gleich/sehr ähnlich. Der *RMSE* sollte möglichst klein sein, da er angibt, in welchem Maß sich die berechneten Werte von den tatsächlichen unterscheiden. Demnach ist für die Vorhersage der austauschbaren Ca-Ionen in Yingtan das *Ordinary Kriging* und das *Universal Kriging/Kriging with external Drift* etwas genauer als das *lineare Regressionsmodell* und das *Inverse-Distance-Weighting*. Da das *Universal Kriging* den besten *RMSE*-Wert liefert, ist dies in diesem Beispiel das beste Interpolationsverfahren.

## Aufgabe 24. Ergebnisexport

a) Exportieren Sie ihr Prognoseergebnis aus Aufg. 23b) im GeoTIFF-Format. (1 Punkte)

```
rgdal_show_exportToProj4_warnings="none"  
library(rgdal)
```

```
## rgdal: version: 1.5-12, (SVN revision 1018)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.0.4, released 2020/01/28
## Path to GDAL shared files: C:/Users/helen/Documents/R/win-library/4.0/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ_VERSION: 631]
## Path to PROJ shared files: C:/Users/helen/Documents/R/win-library/4.0/rgdal/proj
## Linking to sp version:1.4-2
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading rgdal.
## Overwritten PROJ_LIB was C:/Users/helen/Documents/R/win-library/4.0/rgdal/proj
writeGDAL(uk, "data/PrognoseCaexch.tif")
```